

AMENDMENTS TO THE SPECIFICATION

Please replace Paragraph [0016] with the following paragraph rewritten in amendment format:

[0016] Figure 1 illustrates the software architecture of an exemplary service gateway (G) in accordance with the OSGi model. OSGi specification defines common implementation APIs that are suitable for service gateways in a variety of different applications. While the following description is provided with reference to this particular software architecture from the OSGi model, it is readily understood that the broader aspects of the present invention encompass other software architectures as well as other known gateway specifications.

Please replace Paragraph [0017] with the following paragraph rewritten in amendment format:

[0017] In a service gateway environment, applications are designed as a set of services (S), with each service (S) implementing a segment of the overall functionality. These services (S) and other extension services are then packaged into a software container called a bundle (B). In one example, the bundle (B) includes the service 12, which, for example may be any of the following services (S): a HTTP service (S), a Log service (S₂), a device access (S₃), a unified messaging (S₄), a security monitor (S₅), the extension services and/or other suitable services (S₆), as illustrated in Figure 1. The gateway (B) can download the corresponding bundle when it becomes necessary. A ~~The~~ framework (F) handles basic bundle management functionality. The framework (F) may associate with the following other components of the gateway: a

Java Virtual Machine (C₁), an operating system (C₂) and/or hardware (C₃). In various examples, Java Virtual Machine (C₁) may implement Java, JZME or other suitable systems. In one example, the operating system (C₂) may implement a RTOS. In order to share its services (S) with other bundles (B), a the bundle (B) can register any number of services (S) with the framework (F). The An execution thread of a bundle in the framework (F) is referred to as a service instance.

Please replace Paragraph **[0023]** with the following paragraph rewritten in amendment format:

[0023] Referring to Figure 3, a method is provided for managing memory resources in a limited memory environment, where the at least some of the service instances are dependent on other service instances. This method provides one preferred solution to the problem set forth above. The method is initiated Upon receipt of upon receiving a service request 32, ~~the method is initiated at step 32~~, where the service request has a memory space requirement that exceeds the total available memory space associated with the gateway environment. In order to satisfy the service request, the method identifies one or more service instance(s) that are to be terminated from the environment.

Please replace Paragraph **[0025]** with the following paragraph rewritten in amendment format:

[0025] Second, the number of dependent service instances, s^* , is determined at step 36 for each service instance. ~~In an exemplary embodiment, may be~~

~~retrieved from a data store associated with the environment.~~ A The data store generally maintains dependency information for each of the service instances currently resident in the gateway environment. In an exemplary embodiment, the number of dependent services instances may be retrieved from a data store associated with the environment. For each service instance, the data store may include an identifier for a given service instance, the accumulative memory space requirement for the given service instance, and the number of service instances that depend from the given service instance. Although presently preferred, it is envisioned that other techniques may be employed for determining the accumulative memory space requirement and number of dependencies for each service instance.

Please replace Paragraph [0032] with the following paragraph rewritten in amendment format:

[0032] In view of the foregoing, an alternative method is presented for managing memory resources in a limited memory environment, such as a service gateway. Referring to Figure 4, the method is initiated at step 42 upon receiving ~~receipt~~ of a service request. In order to satisfy the service request, the method identifies one or more service instance(s) that are to be deleted from the environment. The technique generally uses dynamic programming.